Juno Delivery Acceptance Review

May 19, 1997

CLCS

Agenda

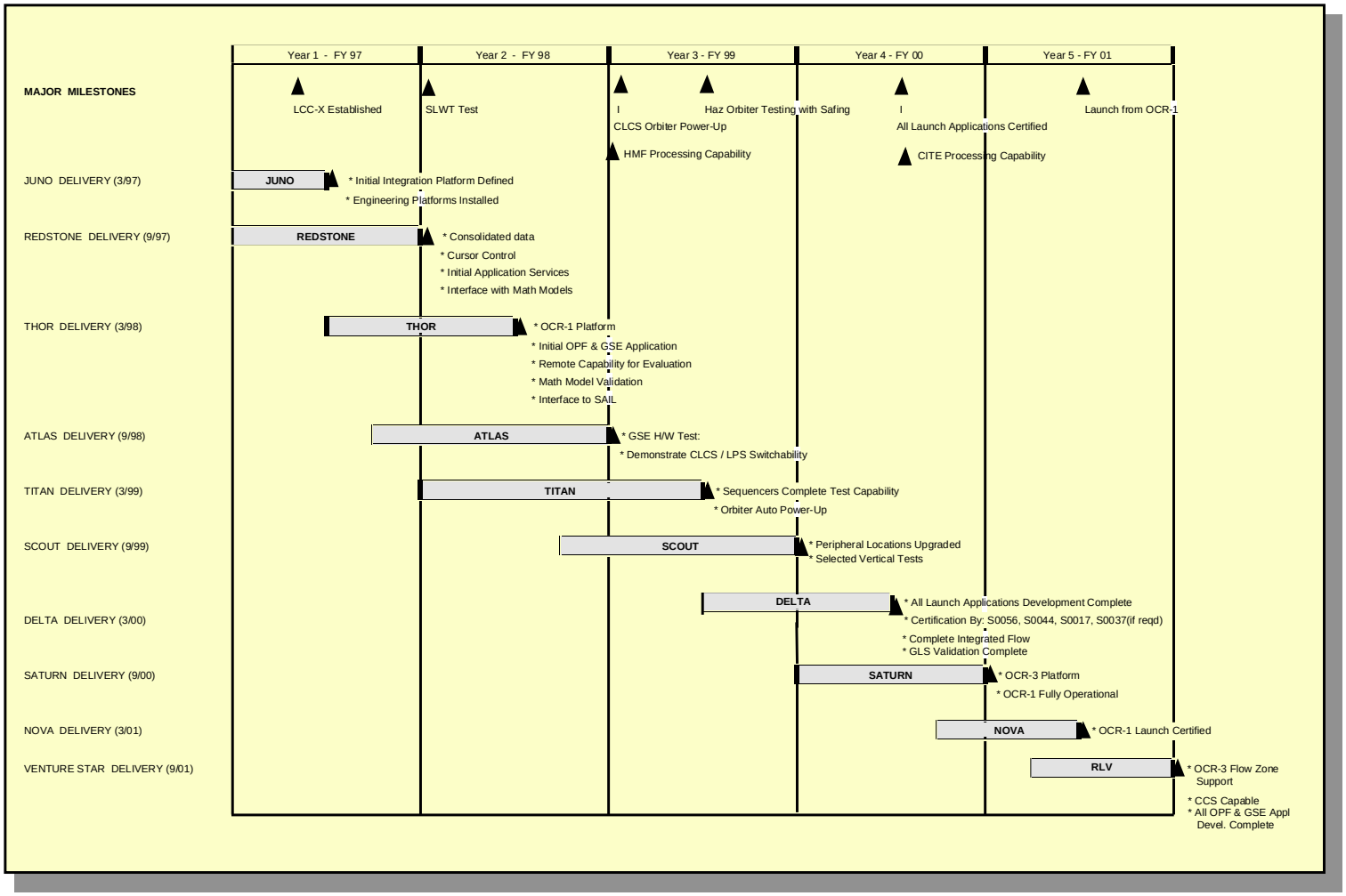
- Purpose and Scope of Delivery Acceptance Review (DAR)
- Summary of Strategic Schedule and Delivery Content
- Summary of Delivered Configuration
- Summary of Delivery Development
- CM Status
- System Integration Summary
- System Test Summary
- Issues and Lessons Learned

Purpose and Scope of DAR

- Purpose
  - A formal meeting where the goal is for the user community to accept a delivery from the development organization
  - Provides a summary of the development, integration and testing of the items to be delivered (including significant known problems)
  - The Juno DAR is intended to standardize the DAR process itself
- Scope
  - Summarize events, system configuration, known problems, interim configurations, specials deals made with the users, etc.
  - This does not completely document a delivery, it points to where the rest of the official documents are

Summary of Strategic Schedule and Delivery Content

- Strategic schedule



Juno Delivery Content

- CSCI's, HWCI's and Threads included in Juno
  - Development Environment Thread
    - COTS hardware and software for the Satellite Development Environments
  - Reliable Messages Thread
    - Custom network software
  - Consolidated SDS Thread
    - Hardware and software for the SDS Gateway
    - SDS' data stream
  - Consolidated Systems Thread
    - Hardware and software for the Consolidated Systems Gateway
- Pathfinders and other significant work
  - LCC-X HCI Testbed
  - MCC Services port
  - Ice Team support

Summary of Delivered Configuration

- H/W diagram for SDE 1

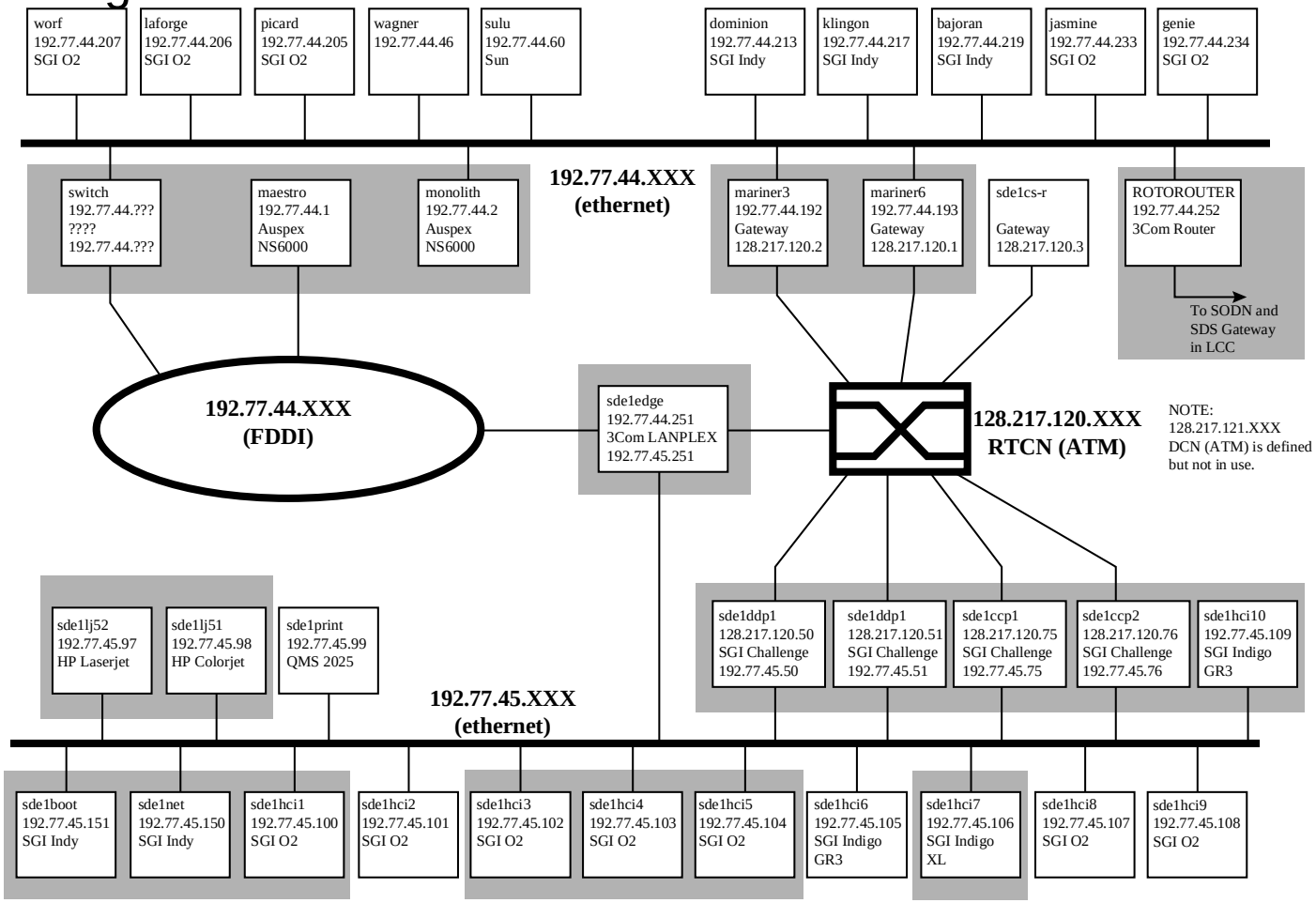
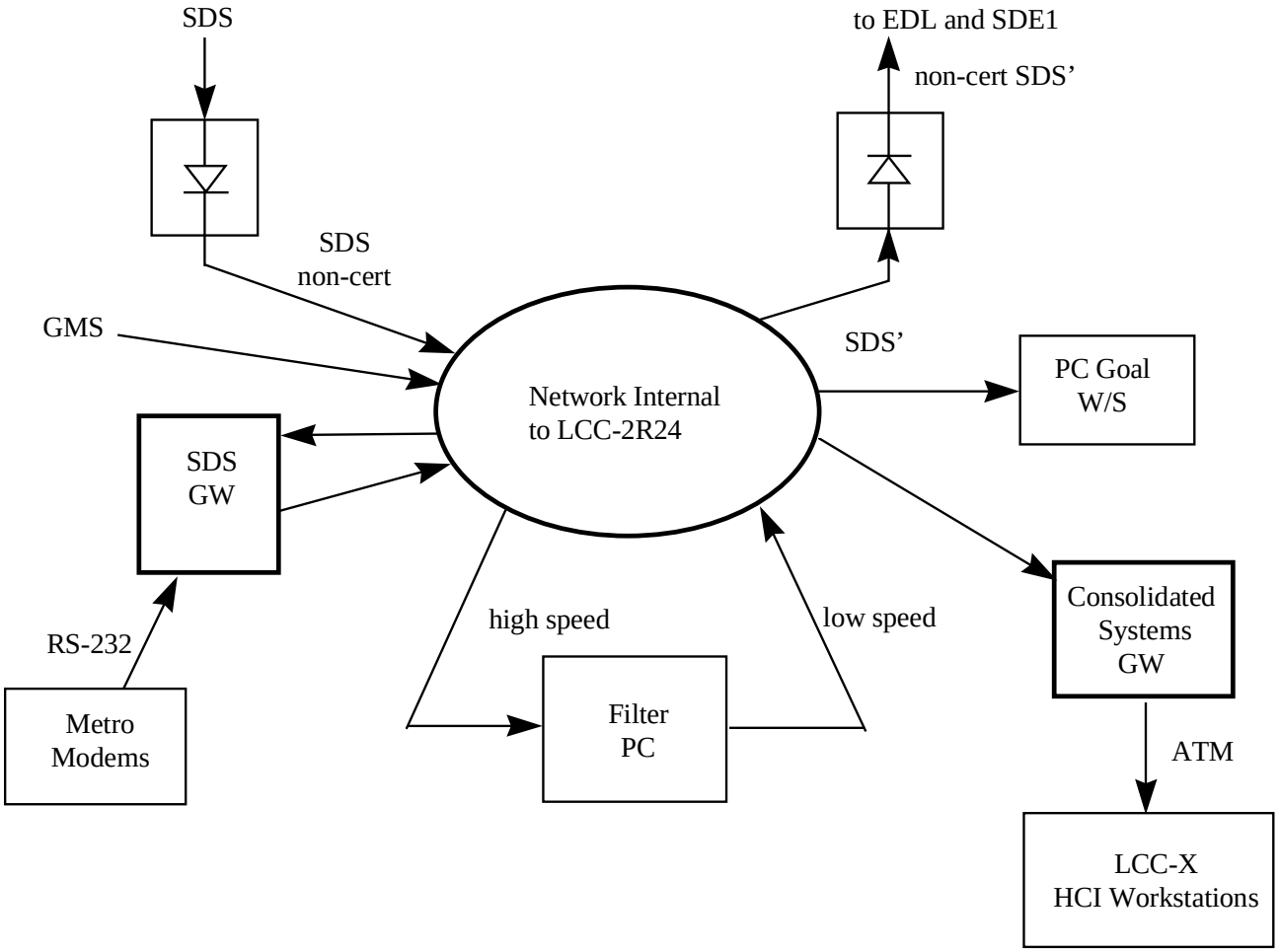


Figure 2: Satelite Development Environment - SDE1
(Shaded areas indicate equipment required for System Test)

Delivered Configuration (cont'd)

LCC Gateway Block Diagram



Delivered Configuration (cont'd)

- S/W architecture diagram
  - Detailed diagram not available
- CM status of CI's
  - Juno software has been backed up to tape and is under CM control
  - Will be converted to automated CM tool when available

Summary of Delivery Development

- Status of Functional Requirements
  - Requirements documented in Juno design panels, baselined in final delivery document
- Status of System Level Specification Requirements
  - Not baselined in Juno time frame
- Summary of CSCI/HWCI Testing (e.g., dates of CIT's)
  - No formal testing of Functional requirements was required or performed
  - Informal testing performed at the Unit level

84K00150-000-01
Rev. Basic

Juno Integration Summary

May 19, 1997

CLCS

Agenda

- Summary
  - SDE-1
  - LCC-X
- Issues
- Lessons Learned

Integration Summary

❒ Summary

The Juno integration effort was intended as a “dry run” to verify integration’s process and procedures for future deliveries. The ability of the integration team to finalize its process and procedures was restricted due to the content of the documentation received at delivery and the lack of a Configuration Management tool. More detailed information can be found in the Juno Integration Report.

- SDE-1
  - MCC services test procedures completed with no critical problems noted (see Issues 01-07, and 12)
  - All basic system and network capabilities were verified
  - Basic end to end data flow capabilities were verified on Indigo2 machines
  - Verified gateways providing Shuttle Data Stream plus metro (SDS’)
- LCC-X
  - Basic end to end data flow capabilities were verified
  - Verified gateways providing Shuttle Data Stream plus metro (SDS’)

Integration Issues

❒ Issues

	Open	Closed
Critical	0	0
Major	8	2
Minor	4	1
Total	**12**	**3**

Integration Issues (cont'd)

Number	CSCI	Criticality	Status	Description
Juno-01	MCC Svc	Major	In Work	System Services global environment variables not sourced each time a xterm is started in Motif Window Manager
Juno-02	MCC Svc	Major	In Work	Event Services: LDS Status/Control not available on remote workstations
Juno-03	MCC Svc	Major	In Work	A core dump occurs when attempting to modify an existing timer
Juno-04	MCC Svc	Major	In Work	Cannot copy an existing timer
Juno-05	MCC Svc	Major	In Work	Deconfigure hangs up on logout
Juno-06	MCC Svc	Minor	In Work	Delog GUI window changes size after application ID added
Juno-07	MCC Svc	Major	In Work	Cannot print man pages
Juno-08	OS	Major	Closed	Cannot find CVS executable in /usr/local/bin to run CM shell procedure
Juno-09	OS	Minor	Closed	The positional .xsession file is being sourced during the login process. This file should not be sourced until ops_cm does it during the configure process.
Juno-10	Reliable Messaging	Major	Closed	Receive error message " error opening msg queue: Permission denied" when attempting to start DDP router process. Attempted to use other ID's (demo, lor_cma) with same result.
Juno-12	MCC Svc	Minor	Open	Remote configuration performed on workstation sde1hci3 from workstation sde1hci1. A remote delog was performed on sde1hci3. The delog file was not found in any user's home directory. The delog was found in directory /clcs/boot/cmtools/Exec. Delogs should not be written into this directory.
Juno-13	User Apps	Minor	Open	Brought up application displays (wind direction, temperature) on indigo workstation. The gr_osview GUI shows that the displays are very CPU intensive.

Integration Lessons Learned

- Lack of a stable configuration during formal integration
  - ATM interface was installed in an o2 workstation after start of integration. Attempts to reboot the workstation failed. SGI technical support was called and a new CPU was installed.
  - System Integration access to LCC-X was impacted by scheduling hardware modifications during designated integration time frame.
  - The SDS gateway in LCC 2R24 is not controlled by system integration or system test personnel. During dry runs the type of data coming out of the gateway was changed (flight 83 to flight 84 to flight 83, etc.). This caused some loss of test time and supports the importance of a stable, frozen configuration for integration and test.
  - Development personnel were using SDE-1 workstations targeted for integration/test personnel during integration activities. Although this caused no problems during Juno integration, it will not be acceptable for future deliveries.

Integration Lessons Learned (cont'd)

- Insufficient documentation
  - Due to schedule constraints, the procedures delivered to system integration did not contain sufficient detail (login ID's)
  - More detailed procedures and configuration documentation will be expected in future deliveries.
- Lack of Configuration Management Tool
  - The CM Shell application of "CVS" was not available on the workstations. This was an addition to the OS baseline, but was not installed because of the configuration freeze for integration and test. OS personnel loaded the application on all workstations at system integration's request.
  - It was discovered late in the integration effort that the same version of **amclm** was not used for all development platforms. In fact, there were three separate versions of **amclm** used: 1 for o2 MCC Services, 1 for reliable messaging, and 1 for gateways. Since no CM capability exists for Juno, it would be very difficult to recreate the **amclm** libraries for each platform. John Porter is working on a plan to baseline **amclm** for the Redstone delivery.

Integration Lessons Learned (cont'd)

- Lack of Directory Structure for Juno
  - The DDP router process would not run unless the user was "root". The resolution of this problem was to modify the permissions on a file in the ddp_router directory to have "write" access by all users (UNIX command chmod 777). Files created/modified by processes need to reside in a directory structure that allows write access, and the creating process needs to assign the correct permissions to the file. System Integration is contacting appropriate personnel to address directory structures and permissions for Redstone.

84K00150-000-01
Rev. Basic

Juno System Test Summary

May 19, 1997

CLCS

Agenda

- System Test Objectives
- Test Summary
- Issues
- Lessons Learned

System Test Objectives

- Test Objectives
  - The Juno System Test was intended to evaluate the baselined CLCS integration and test process.
  - Because the CLCS System Level Specification document was not baselined at test time, no requirements “buy-off” was possible.
    - Quality Assurance did not formally witness and approve the test, though informal support was provided.
  - The test procedures were written in the standard system test format and went to the appropriate level of detail for the items tested.
    - The Juno software is not considered to be completely system tested because there were features, capabilities and programs that were not included in the Juno System Test.
    - Any software or hardware developed for Juno and used in subsequent deliveries will go through formal system testing at that time.

System Test Summary

- System Testing began on April 14th with the execution and completion of 2 test cases
- One test case was executed and completed on April 16th and one on April 22nd.
- Three Issue Reports were generated during the system test, two major and one minor.
- No critical Issue Reports were generated.
- The Juno System Test has been successfully completed.

System Test Issues

- Issues discovered as part of system testing:

	Opened	Closed
Critical	0	0
Major	2	0
Minor	1	0

- Issues discovered during system test as part of system integration:

	Opened	Closed
Critical	0	0
Major	1	0
Minor	1	0

System Test Issues (cont'd)

Number	Title	Opened During	Criticality	Date Opened	Current Status
Juno-11	Telnet from sde1hci1 to sde1ddp-r failed	System Test	Major	4/14/97	Open
Juno-12	Remote delog written into wrong directory	System Integration	Major	4/15/97	Open
Juno-13	Application displays CPU intensive	System Integration	Minor	4/15/97	Open
Juno-14	Telnet to SDS Gateway not working	System Test	Minor	4/22/97	Open
Juno-15	Error received when attempting to start receive process	System Test	Major	4/22/97	Open

System Test Lessons Learned - General

- The configuration management process was not complete, multiple baselines of some software components existed.
- System Integration and System Test dry runs can, in most cases, be performed in parallel
  - System builds and the integration of at least one complete system capability must be done prior to start of test dry runs.
- A CLCS software sustaining process needs to be defined and implemented as soon as possible to allow for:
  - The identification of responsible engineers to handle problem investigation, resolution and functional (CSCI) level re-testing prior to and after a system delivery.
  - Methods for including urgent problem fixes in the current delivery (updating the delivery baseline after start of system test) verses waiting for subsequent deliveries.
  - Better assurance that problems discovered during integration and testing are routed to the appropriate developer in an efficient manner.

System Test Lessons Learned - General (cont'd)

- Requirements buy-off process needs to be refined.
  - Quality Assurance requirements on this process must be clarified.
  - Traceability is required for functional requirements to:
    - CSCI's, HWCI's,
    - CSCI tests and test cases,
    - System Level Specification requirements.
- Hardware identification could be improved. Most workstations were sufficiently labeled, some were not. Labeling should include at least the computer's host name and its network (IP) address.

System Test Lessons Learned - Pre-Test Activities

- Identification of the test configuration was difficult.
  - Hardware diagrams were insufficient
  - Software installation (which programs at which version) on each platform was unknown.
  - There was no definitive source of system configuration information.
- Room access and scheduling was difficult in the LCC-X.
  - The LCC-X facility was not released to the system integration team on schedule. This did not significantly impact testing.
  - Demos for users and non-CLCS people were scheduled in the LCC-X during system integration and testing. Though this did not impact the testing, it had the potential to do so.
  - Not all personnel knew about the system configuration freeze.
  - LCC-X problems will not impact system integration and test in the future, work will be performed in the IDE

System Test Lessons Learned - Pre-Test Activities (cont'd)

- Insufficient documentation was received from CSCI and HWCI developers. This was not unexpected due to the Juno workload and schedule constraints
- Specifically, the following types of documentation will be required in future CLCS deliveries (starting with Redstone):
  - Operating procedures
  - Users' guides
  - CSCI/HWCI test histories and test procedures
  - A Requirements Verification Matrix (for each delivery) which maps functional level test cases to functional level requirements as well as functional level requirements to System Level Specification requirements.
  - Known error conditions (and associated work arounds)
  - Operational dependencies such as:
    - System configuration requirements for CSCI's and HWCI's
    - Data requirements
    - Required interfaces

System Test Lessons Learned - System Test Process

- The system configuration freeze that was to take effect at the start of system integration was not observed.
  - People didn't know the system configuration was frozen.
  - They didn't understand how much of the configuration was frozen and to whom the freeze applied.
  - Changes were made to both hardware and software when they shouldn't have been.
- Clarity of the System Test procedures could be improved.
  - Better identification of the location of steps in a procedure when equipment used in a test case is in multiple locations (for example, test case 3.3 where the LCC-X workstations are on the third floor of the LCC and the SDS gateway is on the second floor).

Other Lessons Learned and Concerns

- Other Lessons Learned
  - The delivery document must be completed prior to the kick-off of subsequent delivery activities
  - The initial goal of using a six month development cycle needs to be updated, delivery work should start nine months ahead of target end date
  - Co-location of developers improves efficiency
  - Pathfinders (e.g., early start console) are beneficial to users, developers and managers
  - Project status reports are only as good as the inputs provided
  - Better dependency management is needed (e.g., S/W, H/W and facilities coordination)
  - All tasks must have a responsible lead/point of contact
- Concerns
  - QA resources are extremely limited
  - Manpower and equipment build-up is slower than anticipated

Appendix A - Deliverable Documents

- List of deliverable documents for Juno, current status (e.g., version #, baselined or in review)
  - Software documents N/A
  - Juno Delivery Document (baselined on 3/31/97)
  - Juno Integration Plan (baselined on 3/24/97)
  - Juno Integration Report (baselined on 4/25/97)
  - Juno System Test Procedures (baselined on 4/14/97)
  - Juno System Test Report (baselined 4/25/97)
- How to get copies of the documents
  - The above documents will be incorporated into the CLCS Documentation CM process
  - Until then, see Mark Taraboletti, Robert Sutton or Dave Reinhardt

Appendix B - Inventory of Delivered Software

- List of CSC's, their version, responsible developer and CM repository info
  - Complete information not available

Appendix C - Inventory of Delivered Hardware

- List of Components, description (model # and configuration), quantity, environment installed in
  - Complete information not available